Variations of the Itai-Rodeh Algorithm for Computing Anonymous Ring Size

Fokkink, Wan; Samsom, Guus

published in
The Art of Modelling Computational Systems: A Journey from Logic and Concurrency to Security and Privacy
2019

DOI (link to publisher)
10.1007/978-3-030-31175-9_1

Variations of the Itai-Rodeh Algorithm for Computing Anonymous Ring Size

Wan Fokkink(✉) and Guus Samsom

Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
w.j.fokkink@vu.nl, guus_samsom@hotmail.com

Abstract. We propose two adaptations of the probabilistic Itai-Rodeh algorithm for computing the size of an anonymous asynchronous ring. This Monte Carlo algorithm (inevitably) allows for wrong outcomes. Our adaptations reduce the chance that this happens. Furthermore, we propose a new algorithm that has a better message complexity.

1 Introduction

In an anonymous network, the nodes do not carry a unique ID. Typically, this is the case if there are no unique hardware IDs (for example, LEGO Mindstorms). Also, when each node has a unique ID but cannot reveal it to the other nodes, this is similar to having no unique IDs at all. For instance, nodes may want to hide such information because of security concerns (e.g. [1]), or because transmitting and storing IDs may be deemed too expensive, as is the case in the IEEE 1394 serial bus [10].

Itai and Rodeh [11] proposed probabilistic distributed algorithms for anonymous rings with asynchronous message-passing communication. One algorithm elects a leader in such networks, another computes its number of nodes. The leader election algorithm is inevitably Las Vegas, meaning that it contains infinite executions in which no leader is ever elected. Moreover, it must require that the nodes know the ring size. The ring size algorithm is inevitably Monte Carlo, meaning that it contains finite executions in which a wrong ring size is computed. The general idea of both algorithms is that nodes repeatedly choose a random ID and then perform an election or ring size algorithm for non-anonymous rings. If a conflict due to identical IDs by different nodes is detected, a new election or size estimation round may be started.

Here we propose some adaptations to the Itai-Rodeh ring size algorithm. First, we let nodes stick to the first random ID they choose, which reduces the chance of a wrong outcome, because each new choice of random IDs may contain an undesirable symmetry that leads to a premature termination of the execution. Second, we prioritize the order in which a node treats incoming messages, to push estimates of the ring size at the nodes upward as quickly as possible. Finally, we propose an alternative ring size algorithm that has a worst-case message complexity of $O(N^2)$, as opposed to the message complexity $O(N^3)$ of the Itai-Rodeh ring size algorithm.

M. S. Alvim et al. (Eds.): Palamidessi Festschrift, LNCS 11760, pp. 3–13, 2019.
https://doi.org/10.1007/978-3-030-31175-9_1

It is with great pleasure that we make this contribution to the Festschrift on the occasion of the 60th birthday of Catuscia Palamidessi. She has made important research contributions on probabilistic systems and anonymity, especially with regard to security and in the context of the π-calculus.

2 Election in Anonymous Rings

We say that a distributed computer network is *anonymous* if its nodes do not carry a unique ID. Some problems that are easily solved in networks with unique node IDs turn out to be insurmountable in anonymous networks. Boldi and Vigna [4] provided effective characterizations of the relations that can be computed on anonymous networks, if a bound on the network size is known.

We first consider election algorithms, which let the nodes in a network choose one leader among them. This first part is included here because it emphasizes the importance of the second part: We will see that precomputing the size of an anonymous ring network plays an important role in electing a leader in such a network. Moreover, several features of the Itai-Rodeh election algorithm, which will be explained below, are carried over to the Itai-Rodeh ring size algorithm, which is the main focus of this paper.

Angluin [2] showed that no election algorithm for anonymous asynchronous networks always terminates. The idea is that if the initial network configuration is symmetric (typically, a ring network in which all nodes are in the same state and all channels are empty), then there is always an infinite execution that cannot escape this symmetry, meaning that no leader is ever elected. Vice versa, if one leader has been elected, all nodes can be given a unique ID using a traversal algorithm (e.g. a depth-first search) initiated by the leader.

In a *probabilistic* algorithm, a node may flip a coin and perform an event based on the outcome of this coin flip. For probabilistic algorithms, one can calculate the probability that an execution from some given set of possible executions will occur. Probabilistic algorithms for which all executions terminate in a correct configuration are in general not so interesting, because any deterministic version of such an algorithm (for example, let the coin flip always yield heads) produces a correct nonprobabilistic algorithm. Therefore generally two classes of probabilistic algorithms are considered. A probabilistic algorithm is *Las Vegas* if the probability that it terminates is greater than 0 and all terminal configurations are correct. It is *Monte Carlo* if it always terminates and the probability that a terminal configuration is correct is greater than 0.

From now on we will consider only anonymous directed ring networks, with asynchronous communication. That is, nodes do not carry a unique ID and the network topology is a ring structure in which messages can only travel in a clockwise direction. Rings have the symmetric topology required for the impossibility results regarding election (explained above) and computing network size (which will be discussed in due course). It is worth noting that for acyclic anonymous networks, an algorithm exists for computing the network size, whereby the execution is started at the leaves of the network and works its way toward the center of the network.

Itai and Rodeh [11] proposed a Las Vegas election algorithm for anonymous directed ring networks in which the probability mass of the infinite executions is 0. In other words, the algorithm terminates with probability 1. (Note that this does allow the presence of infinite executions.) Their algorithm is based on the Chang-Roberts algorithm [5] for non-anonymous rings. Bakhshi, Fokkink, Pang and van de Pol [3] proposed a Las Vegas election algorithm for anonymous rings based on Franklin's algorithm [9] for non-anonymous rings.

In the Itai-Rodeh election algorithm, nodes choose a random ID and then select the node with the largest ID as the leader. Since multiple nodes may choose this largest ID, multiple election rounds may be necessary. In more detail, the algorithm works as follows. Initially, all nodes are active. At the start of each round, the active nodes randomly select an ID and send this ID to their clockwise next neighbor, with a hop counter set to 1. Since nodes are supposed to know the ring size N, a node can recognize from the hop counter when its own message returns after completing a round trip. Each message carries an additional bit that is dirtied if the message visits a node with the same ID that is not its originator.

Passive nodes simply pass on incoming messages, with the hop counter increased by 1. An active node that receives a message (for its current round) compares the ID of the message with its own randomly chosen ID for this round. There are three cases:

- If the message ID is smaller than the node ID, the message is purged.
- If the message ID is larger than the node ID, the node becomes passive and the message is passed on, with the hop counter increased by 1.
- If the message ID is equal to the node ID but its hop counter is smaller than N, then the message is passed on, again with the hop counter increased by 1, but also with a dirtied bit.

If a message returns to its originator with the hop counter N, the recipient checks whether the bit still clean. If so, the node becomes the leader (and it is certain that all other nodes are by now passive). If on the other hand the bit has been dirtied, then the node proceeds to a next election round (because another node chose the same ID in this round) and chooses a new random ID.

Messages carry the round number of the sender to avoid confusion, in case a message of an earlier round reaches its destination after some delay. (A variant of the algorithm without round numbers, in the case of FIFO channels, was proposed in [7,8].)

3 Computing the Size of an Anonymous Ring

As mentioned above, in the Itai-Rodeh election algorithm it is required that nodes know the ring size. This requirement is crucial, because there is no Las Vegas algorithm to compute the size of anonymous rings; every probabilistic algorithm for computing the size of an anonymous ring must allow for incorrect outcomes. This implies that there is no Las Vegas algorithm for election in anonymous rings if nodes do not know the ring size, because when there is one leader, network size can be computed using a traversal algorithm initiated by the leader.

The proof that there is no Las Vegas algorithm to compute the size of an anonymous ring goes roughly as follows. Suppose that such an algorithm does exist. We apply it on an anonymous ring of size $N > 2$. Consider an execution E that terminates with the correct outcome N. We cut the ring open between two of its nodes, make a copy of the resulting line of nodes, and glue the two parts together, yielding an anonymous ring of size $2N$. Now we can perform the execution E twice, on the two halves of this ring. Hereby it is crucial that at the two places where the two halves were glued together, the recipient of a message cannot recognize that the message now originates from a node in the other half, due to anonymity. This execution on the ring of size $2N$ terminates with the incorrect outcome N.

The Itai-Rodeh ring size algorithm targets anonymous directed rings. We have seen that it must be a Monte Carlo algorithm, meaning that it must allow for incorrect outcomes. However, in the Itai-Rodeh ring size algorithm, the probability of an erroneous outcome can be arbitrarily close to 0 by letting the nodes randomly select IDs from a sufficiently large domain.

Each node p maintains an estimate est_p of the ring size; initially $est_p = 2$. During any execution of the algorithm, est_p will never exceed the correct estimate N. The algorithm proceeds in estimate rounds. Every time a node finds that its estimate is too conservative, it moves to another round. That is, each node p initiates an estimate round at the start of the algorithm as well as at every update of est_p. The following detailed description of the algorithm is based on the ones in [6, 12].

In each round, p randomly selects an ID id_p from $\{1, \ldots, R\}$ for some positive number R and sends the message $(est_p, id_p, 1)$ to its next neighbor. The third value is a hop count, which is increased by 1 every time the message is forwarded.

Now p waits for a message (est, id, h) to arrive. An invariant of such messages is that always $h \leq est$. When a message arrives, p acts as follows, depending on the parameter values in this message:

- $est < est_p$:
  The estimate of the message is more conservative than p's estimate, so p dismisses the message.
- $est > est_p$:
  The estimate of the message improves on p's estimate, so p increases its estimate. We distinguish between two cases:
  - $h < est$:
    The estimate est may be correct. So p sends $(est, id, h + 1)$ to give the message the chance to complete its round trip. Moreover, p performs $est_p \leftarrow est$.
  - $h = est$:
    The estimate est is too conservative because the message traveled est hops but did not complete its round trip. Therefore p performs $est_p \leftarrow est + 1$.

– $est = est_p$:
  The estimate of the message and that of p agree. We distinguish between two cases:
  • $h < est$:
    p sends $(est, id, h+1)$ to give the message the chance to complete its round trip.
  • $h = est$:
    We again distinguish between two cases:
    ∗ $id \neq id_p$:
    The estimate est is too conservative because the message traveled est hops but did not complete its round trip. Therefore p performs $est_p \leftarrow est+1$.
    ∗ $id = id_p$:
    Possibly p's own message returned (or a message originating from another node est hops before p that unfortunately happened to select the same ID as p in this estimate round). In this case, p dismisses the message.

When the algorithm terminates, $est_p \leq N$ for all nodes p, because a node increases its estimate only when it is certain that its current estimate is too conservative. Furthermore, est_p converges to the same value at all nodes p. If this were not the case, clearly there would be nodes p and q where p is q's predecessor in the ring and p's final estimate is larger than that of q. But then p's message in its final estimate round would have increased q's estimate to p's estimate.

The Itai-Rodeh ring size algorithm is a Monte Carlo algorithm: it may terminate with an estimate smaller than N. This can happen if in a round with an estimate $est < N$ all nodes at distance est from each other happen to select the same ID. The probability that the algorithm terminates with an incorrect outcome clearly becomes smaller when the domain $\{1, \ldots, R\}$ from which random IDs are drawn is made larger. This probability tends to 0 when R tends to infinity, for a fixed N.

In particular, if N is a prime number, it is not hard to compute the probability of a correct outcome, under the simplifying assumption that nodes never skip an estimate round. Since N is prime, the nodes only terminate with an estimate $1 < est < N$ if they have all chosen the same identity. Hence, the correct ring size N is computed if in each election round $e = 2, \ldots, N-1$, the nodes do not all select the same identity. In other words, the probability that ring size N is computed is

$$\left(1 - \frac{1}{R^{N-1}}\right)^{N-2}$$

The side condition that we only consider executions in which no node skips a round is essential here (and means the probability above is in fact a bit too conservative). Namely, a node p skips an estimate round if it receives a message (est, id, h) with either $h = est = est_p + 1$ or $est > est_p + 1$. In those cases p skips (at least) the estimate round $est_p + 1$.

The worst-case message complexity of the Itai-Rodeh ring size algorithm is $O(N^3)$: each node starts at most $N-1$ estimate rounds, and during each round it sends out one message, which takes at most N steps. The worst-case time complexity is $O(N^2)$, under the assumption that messages take at most one time unit to reach their destination, since then the ith estimate round completes after at most $i \cdot N$ time units, for $i = 1, \ldots, N-1$.

4 Adaptations of the Itai-Rodeh Ring Size Algorithm

We now propose some adaptations of the Itai-Rodeh ring size algorithm, with the aims to increase the chance of a correct outcome and to decrease its message complexity. We have also performed some simulations of implementations of the original Itai-Rodeh ring size algorithm and our adaptations, for small values of N, to get an impression of the impact of the adaptations.

4.1 Choose a Random ID Only Once

A first, simple adaptation of the Itai-Rodeh ring size algorithm is to let the nodes stick to the first random ID they choose, instead of selecting a new random ID at each increased estimate. That this is beneficial for the chance of a correct outcome is clear in the case that N is prime. In that case the only possibility for a wrong outcome is if all nodes choose the same ID at the start. In other words, the probability that ring size N is computed is

$$1 - \frac{1}{R^{N-1}}$$

Basically, the principle that is essential for the Itai-Rodeh election algorithm, letting a node choose a new random ID at each round, is harmful for the Itai-Rodeh ring size algorithm. The latter algorithm may terminate with a wrong outcome if the chosen IDs in an estimate round display a certain symmetry. Letting the nodes choose a new ID at each round increases the chance that in some round the chosen IDs contain a detrimental symmetry.

4.2 Prioritization of Received Messages

For simplicity, the analysis of the probability that the correct ring size is computed, at the end of Sect. 3, assumed that nodes never skip an estimate round. Since estimations of the ring size are guaranteed to be conservative, the possibility that a node skips an estimate round increases the chance that an execution of the Itai-Rodeh ring size algorithm terminates with the correct outcome. To increase the chance that estimate rounds are skipped, it is beneficial to define a prioritization on the order in which a node p treats concurrently received messages in its buffer.

- First of all, the priority is based on the value of est.
- Second, the priority is based on the value of h.
- Third (so if two received messages carry the same estimate and hop counter), a message with an identity different from id_p has a higher priority than a message with the identity id_p.

In all three cases the priority aims to increase the chance that est_p skips a value.

4.3 Each Node Sends Out Only One Message

To decrease the worst-case message complexity from $O(N^3)$ to $O(N^2)$, we let each node send out only one message, which in the worst case (from the point of message complexity) completes the entire round trip. We exclude the value est_p of the sender p from the message, so that it only contains the ID of p and a hop count. The intention is that each of these messages completes the round trip and helps to increase estimates of visited nodes on the way.

Again, initially $est_p = 2$ at each node p, and est_p will never exceed N. Each node p randomly selects an ID id_p from $\{1, \ldots, R\}$ for some positive number R and sends the message $(id_p, 1)$ to its next neighbor. Now p waits for a message (id, h) to arrive.

When p receives a message $m = (id_p, h)$ with $h \geq est_p$, it assumes that this message originates from p itself, so (only) in that case the received message is not forwarded, while est_p is updated to h. If p later receives another message showing that the value $est_p = h$ is too conservative, then p must forward the message m after all. Therefore a Boolean variable $passive_p$ is set when p receives m, to recall that p must send the message $(id_p, est_p + 1)$ if the value of est_p is increased at some later moment in time. Initially the value of $passive_p$ is *false*.

When node p receives a message (id, h), it acts as follows, depending on the parameter values in this message:

- $h < est_p$:
  p forwards the message with the hop count increased by 1, i.e., $(id, h + 1)$.
- $h \geq est_p$:
  If $passive_p = true$, then p forwards $(id_p, est_p{+}1)$, because p earlier stopped the incoming message (id_p, est_p) by mistake. (We note that in this case $h > est_p$, because a node never receives two different messages with the same hop count.) Moreover, p then performs $passive_p \leftarrow false$. We distinguish between two cases:
  * $id \neq id_p$:
  Since the message does not originate from p, the value of est_p is too conservative. Therefore p performs $est_p \leftarrow h + 1$. Furthermore, it forwards $(id, h + 1)$.
  * $id = id_p$:
  Possibly p's own message returned (or a message originating from another node h hops before p that unfortunately happened to select the same ID as p). Then p performs $est_p \leftarrow h$. Moreover, p performs $passive_p \leftarrow true$, to recall that it did not forward the message (id_p, est_p).

Since no message can be forwarded beyond the node it originated from, it is not hard to see that the algorithm terminates, and that then $est_p \leq N$ at all nodes p. Moreover, the $passive_p$ flags ensure that each node at any moment has stopped at most one message. Since each node sends out one message, this implies that each node stops exactly one message permanently. Clearly, the IDs of a node p and of the message m that is stopped permanently at p coincide, and the final value of est_p coincides with the value of the hop counter of m. The fact that the predecessor q of p in the ring forwarded m implies that upon termination, $est_q \geq est_p$. Since this inequality holds for each pair of neighbors in the ring, it follows that upon termination, all nodes carry the same estimate.

This algorithm has worst-case message complexity $O(N^2)$ (compared to $O(N^3)$ for the Itai-Rodeh ring size algorithm), because each node sends out only one message, which takes at most N steps.

4.4 Implementation and Simulation Results

We implemented in Java the original Itai-Rodeh ring size algorithm, as well as the two adaptations proposed in Sects. 4.1 and 4.2, and the following pseudocode description of our algorithm proposed in Sect. 4.3 which describes how a node p acts when it receives a message (id, h). (Initially est_p has the value 2, $passive_p$ has the value *false*, and p sends the message $(id_p, 1)$ to its successor $next_p$ in the ring.)

```
if h < est_p then
│  send (id, h + 1) to next_p;
else
│  if passive_p = true then
│  │  send (id_p, est_p + 1) to next_p;
│  │  passive_p ← false;
│  end
│  if id ≠ id_p then
│  │  est_p ← h + 1;
│  │  send (id, h + 1) to next_p;
│  else
│  │  est_p ← h;
│  │  passive_p ← true;
│  end
```

We ran a million simulations for both $N = 12$ and $N = 13$, to get an impression of the impact of the adaptations on the performance. These numbers were chosen because 12 has relatively many divisors while 13 is prime. The outcomes of these experiments with regard to the Itai-Rodeh ring size algorithm and the adaptations proposed in Sects. 4.1 and 4.2 are plotted in Fig. 1. The horizontal axis charts the possible ring sizes, ranging from 2 up to and including $N-1$, while the vertical axis expresses how many of the experiments produced a certain ring size; in the plot for $N = 12$, these numbers must be multiplied with 10^4. On the horizontal axis, the correct ring size N has been excluded because this column would dwarf the other ones. Already with these small values for N, the simulations took a significant amount of time, due to the message complexity of $O(N^3)$.

For $N = 12$ the impact of sticking to the first chosen ID is already significant, and for $N = 13$ it makes the probability of computing a size between 2 and

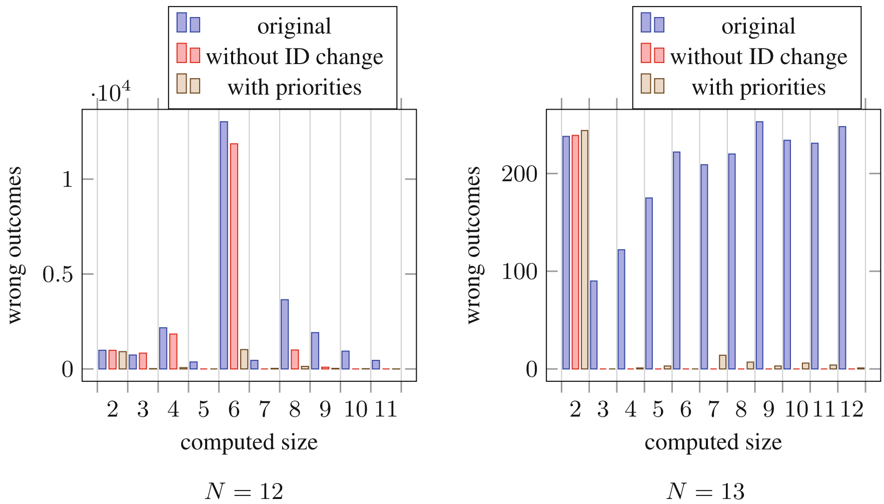
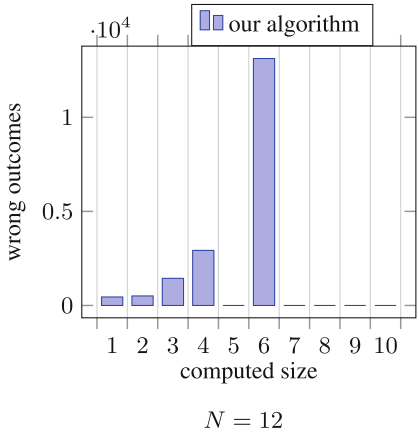


Fig. 1. Simulation results for three variants of the Itai-Rodeh ring size algorithm, with $N = 12$ and $N = 13$.

13 drop to 0, because 13 is prime. The impact of imposing priorities on the order in which concurrently received messages are treated is rather dramatic in our experiments. (In this implementation we included ID changes, to clearly distinguish the effects the two optimizations have on the performance of the Itai-Rodeh algorithm.) This dramatic effect is due to the fact that in our simulations, channel delays were chosen to be negligible, so that in-buffers at nodes will often contain multiple messages. In case of a larger channel delay, in-buffers will mostly contain no more than one message, in which case clearly prioritization has no effect at all. Finally, we note that for $N = 13$, the dip in wrong outcomes for the values 3, 4, and 5 for the original Itai-Rodeh algorithm is caused by the fact that nodes may skip estimate rounds, as explained at the end of Sect. 3. The chance that this happens is larger for small values of est_p, because then it is more likely that a message (est, id, h) arrives at p with $h = est = est_p + 1$ or $est > est_p + 1$.

In Fig. 2 the simulation results of our ring size algorithm from Sect. 4.3 are plotted, again for $N = 12$ and $N = 13$. For the sake of a clear comparison with the original Itai-Rodeh algorithm, we refrained from using prioritization of received messages. It can be observed that the probability of computing a wrong outcome is comparable to the original Itai-Rodeh algorithm without ID changes. However, these simulations took much less time, owing to the message complexity of $O(N^2)$.



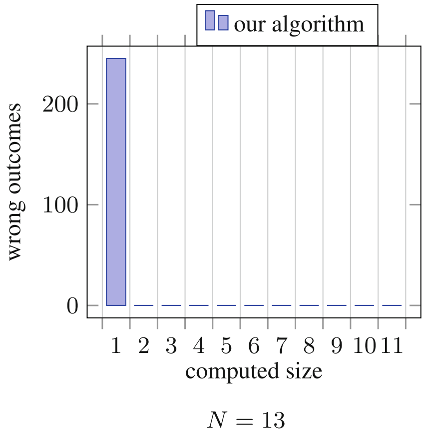


Fig. 2. Simulation results for our ring size algorithm, with $N = 12$ and $N = 13$.

5 Conclusion

We proposed two optimizations of the Itai-Rodeh algorithm for computing the size of an anonymous ring: nodes stick to the first random ID they select, and the treatment of received messages is prioritized to stimulate the fast propagation of larger estimates through the ring. Furthermore, we proposed a new algorithm for computing the size of an anonymous ring, in which each node sends out only one message, so that the worst-case message complexity is $O(N^2)$ (compared to $O(N^3)$ for the Itai-Rodeh algorithm).

The implementations and the simulation experiments are available at https://github.com/gsamsom/Itai-Rodeh-simulatie.

In [8], the probabilistic model checker PRISM was used to analyze a finite-state version of the Itai-Rodeh leader election algorithm. Moreover, this algorithm served as a benchmark for different optimization techniques for probabilistic model checking, e.g. in [13]. Likewise, to complement our simulation results, it would be interesting to apply probabilistic model checking to analyze the probability that the different variations of the Itai-Rodeh ring size algorithm terminate correctly, on anonymous rings of varying sizes. Moreover, these variations could be employed for a benchmark in probabilistic model checking.

References

1. Andrés, M., Palamidessi, C., van Rossum, P., Sokolova, A.: Information hiding in probabilistic concurrent systems. Theoret. Comput. Sci. **412**(28), 3072–3089 (2011)
2. Angluin, D.: Local and global properties in networks of processors. In: Proceedings of the STOC 1980, pp. 82–93. ACM (1980)

3. Bakhshi, R., Fokkink, W., Pang, J., van de Pol, J.: Leader election in anonymous rings: Franklin goes probabilistic. In: Proceedings of the IFIP TCS 2008, pp. 57–72. IFIP (2008)
4. Boldi, P., Vigna, S.: An effective characterization of computability in anonymous networks. In: Welch, J. (ed.) DISC 2001. LNCS, vol. 2180, pp. 33–47. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45414-4_3
5. Chang, E., Roberts, R.: An improved algorithm for decentralized extrema-finding in circular configurations of processes. Commun. ACM **22**, 281–283 (1979)
6. Fokkink, W.: Distributed Algorithms: An Intuitive Approach. MIT Press, Cambridge (2013)
7. Fokkink, W., Pang, J.: Simplifying Itai-Rodeh leader election for anonymous rings. In: Proceedings of the AVOCS 2004. Electronic Notes in Theoretical Computer Science, vol. 128, no. 6, pp. 53–68. Elsevier (2005)
8. Fokkink, W., Pang, J.: Variations on Itai-Rodeh leader election for anonymous rings and their analysis in PRISM. J. Univ. Comput. Sci. **12**, 981–1006 (2006)
9. Franklin, W.R.: On an improved algorithm for decentralized extrema-finding in circular configurations of processes. Commun. ACM **25**(5), 336–337 (1982)
10. IEEE Computer Society: IEEE standard for a high performance serial bus. Technical report, IEEE (1994)
11. Itai, A., Rodeh, M.: Symmetry breaking in distributed networks. Inf. Comput. **88**(1), 60–87 (1990)
12. Tel, G.: Introduction to Distributed Algorithms, 2nd edn. Cambridge University Press, Cambridge (2000)
13. Timmer, M., Katoen, J.P., van de Pol, J., Stoelinga, M.: Confluence reduction for Markov automata. Theoret. Comput. Sci. **655**, 193–219 (2016)